



Google Developer Student Clubs  
Soongsil University

# GDSC Soongsil NFF Study

## 2. Vue Code-jam

GDSC Soongsil 이강현 공소나

```
  enOrg : filterByOrg ? study lead_organization === filterByOrg : true
  !status = filterByStatus ? study.status === filterByStatus : true
  matchStatus) {
    ...
  }
}

function filterStudies({ studies, filterByOrg = false, filterByStatus = false }) {
  return studies.filter(study => {
    if (filterByOrg) {
      study lead_organization === filterByOrg
    } else if (filterByStatus) {
      study status === filterByStatus
    } else {
      study lead_organization === filterByOrg || study status === filterByStatus
    }
  })
}
```





# Vue

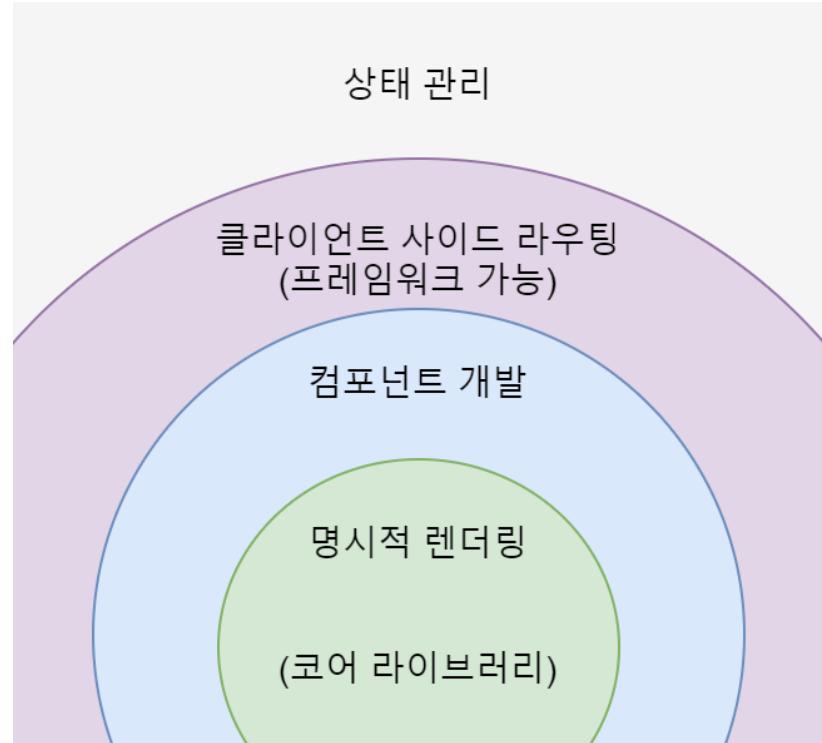
사용자 인터페이스 구축을 위한 JavaScript 프레임워크

인터페이스를 효율적으로 개발할 수 있는 컴포넌트 기반 프로그래밍 모델을 제공



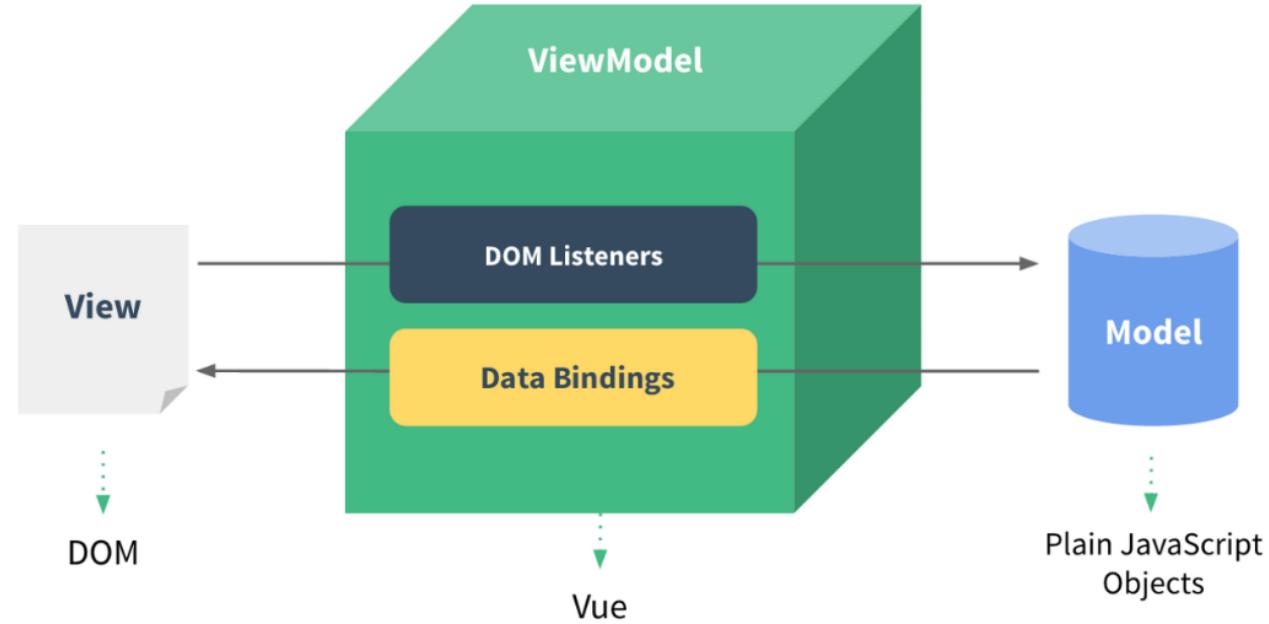
# 프로그레시브 자바스크립트 프레임워크

웹 사용자 인터페이스를 만들기 위한 쉽고 강력하며 다재다능한 프레임워크

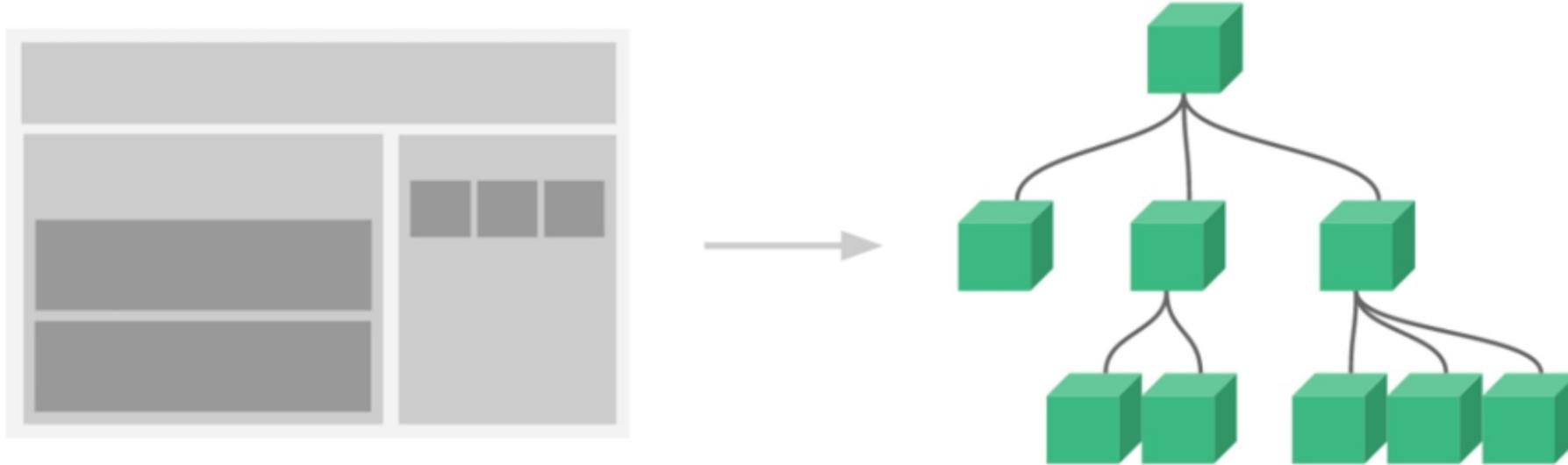


- 빌드 과정 없이 정적 HTML에 적용
- 모든 페이지에 웹 컴포넌트로 추가
- 싱글 페이지 어플리케이션 (SPA: Single-Page Application)
- Fullstack / 서버 사이드 렌더링 (SSR: Server-Side-Rendering)
- Jamstack / 정적 사이트 생성 (SSG: Static-Site-Generation)
- 데스크톱, 모바일, WebGL 또는 터미널을 대상으로 하는 경우

# MVVM 패턴



## 컴포넌트 기반

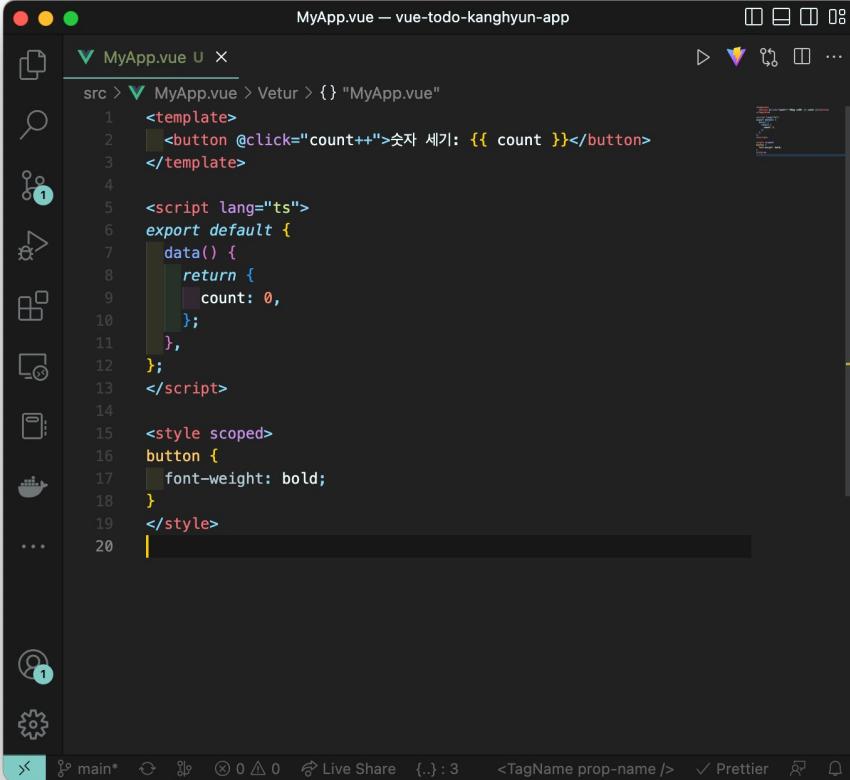


# 리액트와 앵귤러의 장점을 가진 프레임워크

	Angular	React	Vue
언어	TypeScript	JSX	ES6
Virtual DOM	x	o	o
장점	구조화가 잘 되어있어 유지보수 용이, 구글의 지원	다양한 이벤트 핸들링에 용이, 페이스북의 지원, 다양한 플랫폼 개발 가능	낮은 진입장벽, Angular2와 React의 장점 적용
단점	높은 러닝커브	필수 라이브러리가 많음, JSX 한곳에 로직구현	상대적으로 레퍼런스가 적음
적합한 환경	엔터프라이즈	많은 이벤트 사용	빠른 개발속도가 필요하고 많은 교육이 어려울 때

양방향 데이터 바인딩을 지원한다는 점에서 Angular 와 공통점을 가지고 있고  
단방향 데이터 흐름 방식을 취하고 가상 DOM을 지원한다는 점에서 리액트와 공통점을 가지고 있다.

# 선언적 렌더링과 반응성

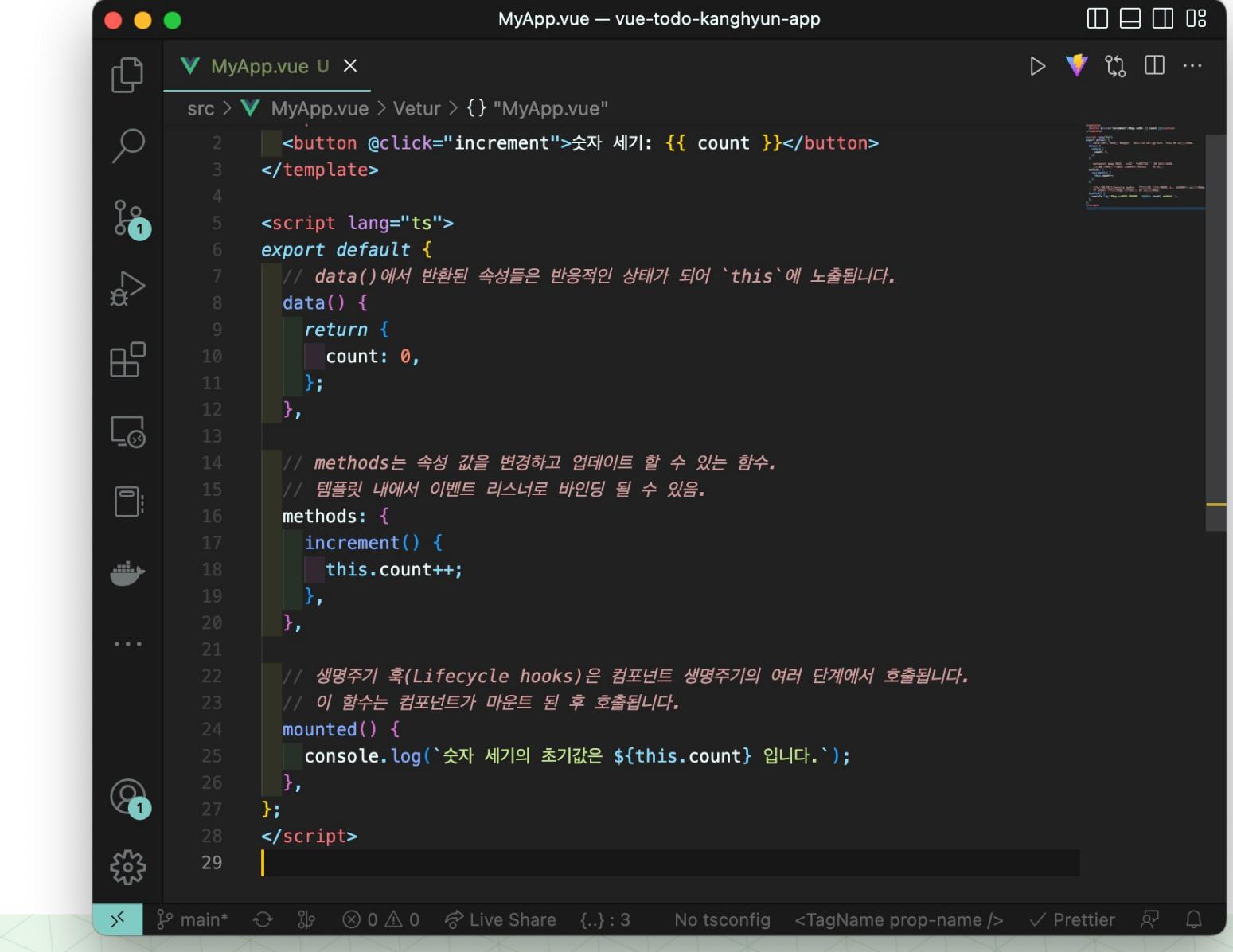


```
MyApp.vue — vue-todo-kanghyun-app
src > MyApp.vue > Vetur > {} "MyApp.vue"
1  <template>
2    <button @click="count++">숫자 세기: {{ count }}</button>
3  </template>
4
5  <script lang="ts">
6    export default {
7      data() {
8        return {
9          count: 0,
10        };
11      },
12    };
13  </script>
14
15  <style scoped>
16    button {
17      font-weight: bold;
18    }
19  </style>
20
```

- SFC (Single File Component)
- 컴포넌트의 논리(Javascript), 템플릿(HTML) 및 스타일(CSS)를 하나의 파일에 캡슐화
- **.vue**
- 반응성
- State 변경을 추적하고, 변경이 발생하면 효율적인 DOM 업데이트를 자동으로 수행

# 옵션 API vs 컴포지션 API

# 옵션 API



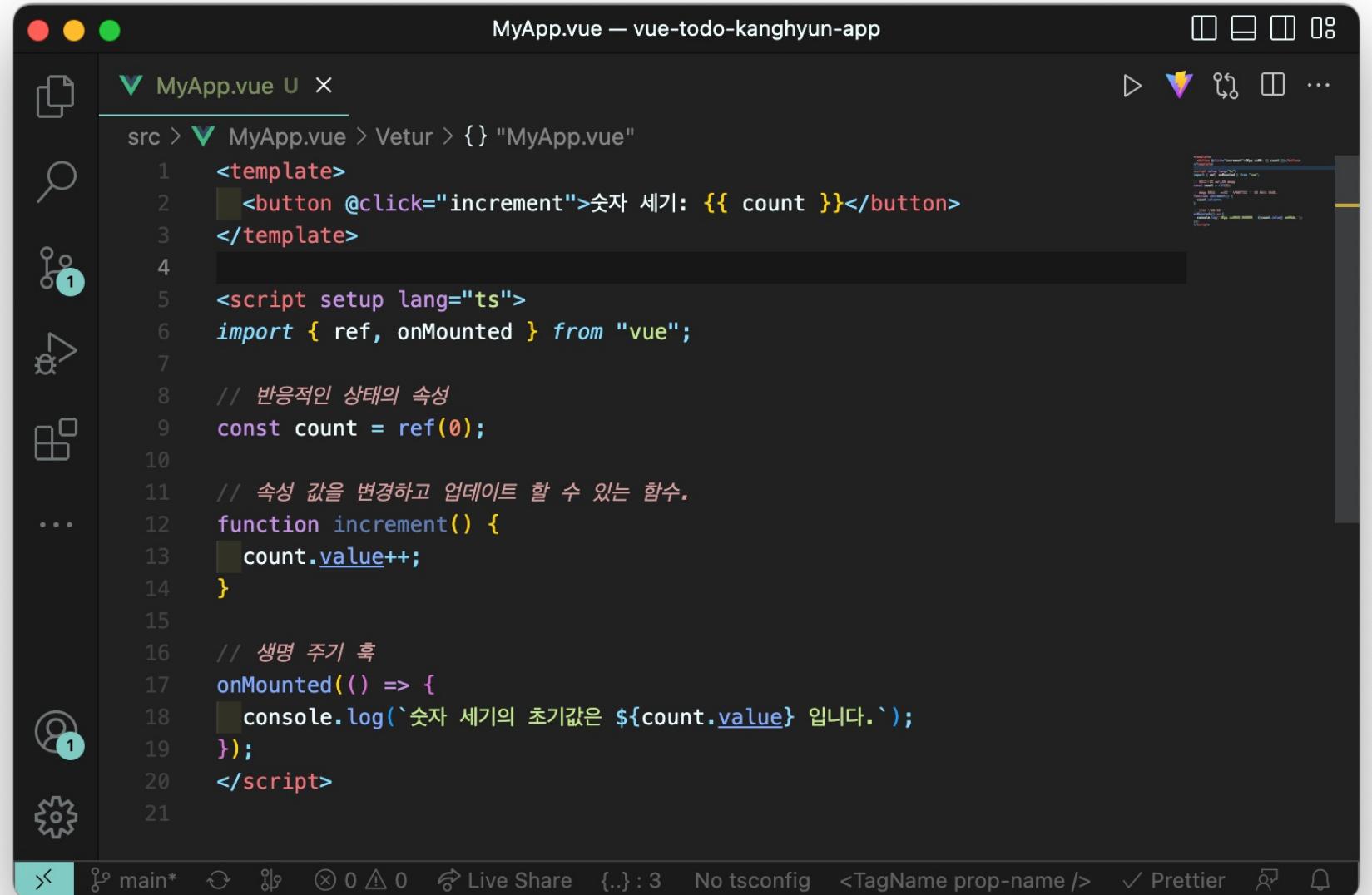
The screenshot shows a code editor window with the file `MyApp.vue` open. The code is written in Vue.js using the Options API. The editor has a dark theme with syntax highlighting. A sidebar on the left contains icons for file operations like copy, search, and refresh, along with a notifications icon showing one notification.

```
<button @click="increment">숫자 세기: {{ count }}</button>
</template>

<script lang="ts">
export default {
  // data()에서 반환된 속성들은 반응적인 상태가 되어 `this`에 노출됩니다.
  data() {
    return {
      count: 0,
    };
  },
  // methods는 속성 값을 변경하고 업데이트 할 수 있는 함수,
  // 템플릿 내에서 이벤트 리스너로 바인딩 될 수 있음.
  methods: {
    increment() {
      this.count++;
    },
  },
  // 생명주기 흑(Lifecycle hooks)은 컴포넌트 생명주기의 여러 단계에서 호출됩니다.
  // 이 함수는 컴포넌트가 마운트 된 후 호출됩니다.
  mounted() {
    console.log(`숫자 세기의 초기값은 ${this.count} 입니다.`);
  },
};
</script>
```

The code includes explanatory comments in Korean. It defines a button with a click event that increments a counter. The counter is initialized to 0 in the `data()` method. The `increment()` method increments the counter by 1. The `mounted()` lifecycle hook logs the initial value of the counter to the console.

# 컴포지션 API



The screenshot shows a code editor window for a file named `MyApp.vue`. The code demonstrates the Composition API in Vue.js. It includes a template with a button that increments a counter, and a script block with logic for state management and lifecycle hooks.

```
src > MyApp.vue > Vetur > {} "MyApp.vue"
1  <template>
2    <button @click="increment">숫자 세기: {{ count }}</button>
3  </template>
4
5  <script setup lang="ts">
6    import { ref, onMounted } from "vue";
7
8    // 반응적인 상태의 속성
9    const count = ref(0);
10
11   // 속성 값을 변경하고 업데이트 할 수 있는 함수.
12   function increment() {
13     count.value++;
14   }
15
16   // 생명 주기 흔
17   onMounted(() => {
18     console.log(`숫자 세기의 초기값은 ${count.value} 입니다.`);
19   });
20 </script>
21
```

# 디렉티브

v- 접두사로 시작하는 특수한 속성으로 Vue 템플릿 문법

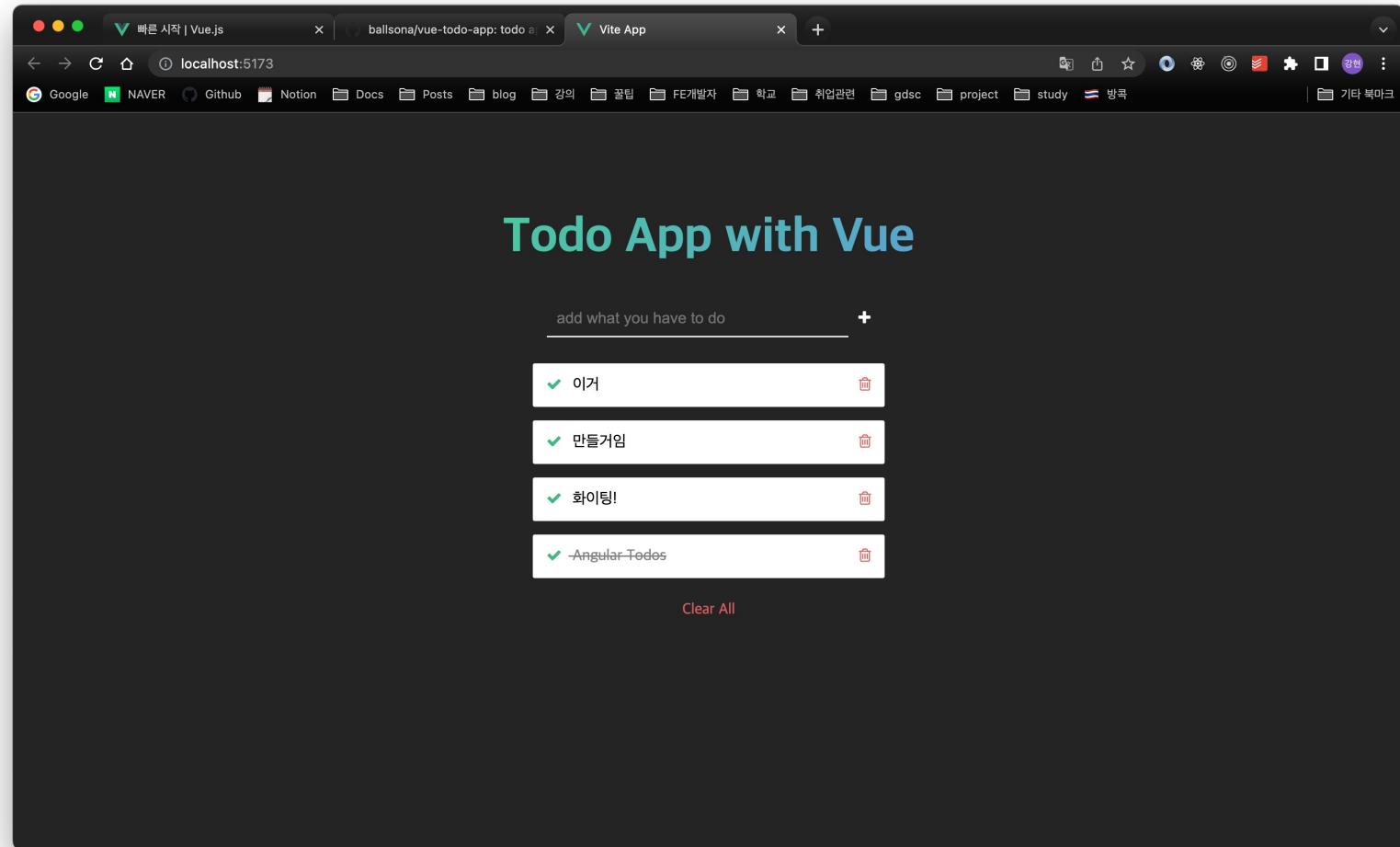
▼ 빌트인 디렉티브

- v-text
- v-html
- v-show
- v-if, v-else, v-else-if
- v-for
- v-on `@` : DOM 이벤트 바인딩
- v-bind `:` : HTML 속성을 반응적으로 업데이트
- v-model : 양방향 바인딩
- v-slot `#`
- v-pre
- v-once
- v-memo
- v-cloak



# Hello Vue!







```
kanghyun@KangHyun-MacBook:~/projects/vue-todo-app ⌘1
Last login: Sun Jan 15 20:15:15 on ttys000
cd projects
~/projects> npm init vue@latest
✓ 22:24:52
✓ 22:25:04

Vue.js – The Progressive JavaScript Framework

✓ Project name: vue-todo-app
✓ Add TypeScript? No Yes
✓ Add JSX Support? No Yes
✓ Add Vue Router for Single Page Application development? No Yes
✓ Add Pinia for state management? No Yes
✓ Add Vitest for Unit Testing? No Yes
✓ Add an End-to-End Testing Solution? No
✓ Add ESLint for code quality? No Yes
✓ Add Prettier for code formatting? No Yes

Scaffolding project in /Users/kanghyun/projects/vue-todo-app...

Done. Now run:

cd vue-todo-app
npm install
npm run lint
npm run dev
```

# With Router

The screenshot shows a Vue.js application structure in VS Code. The project is named "VUE-TODO-ROUTER-APP". The file "App.vue" is open in the editor, displaying the following code:

```
src > App.vue > Vetur > {} "App.vue" > template > RouterView
1 <script setup lang="ts">
2 import { RouterLink, RouterView } from "vue-router";
3 import HelloWorld from "./components/HelloWorld.vue"; File '/Users
4 </script>

<template>
<header>


<div class="wrapper">
<HelloWorld msg="You did it!" />

<nav>
<RouterLink to="/">Home</RouterLink>
<RouterLink to="/todo">Todo</RouterLink>
</nav>
</div>
</header>

<RouterView />
</template>

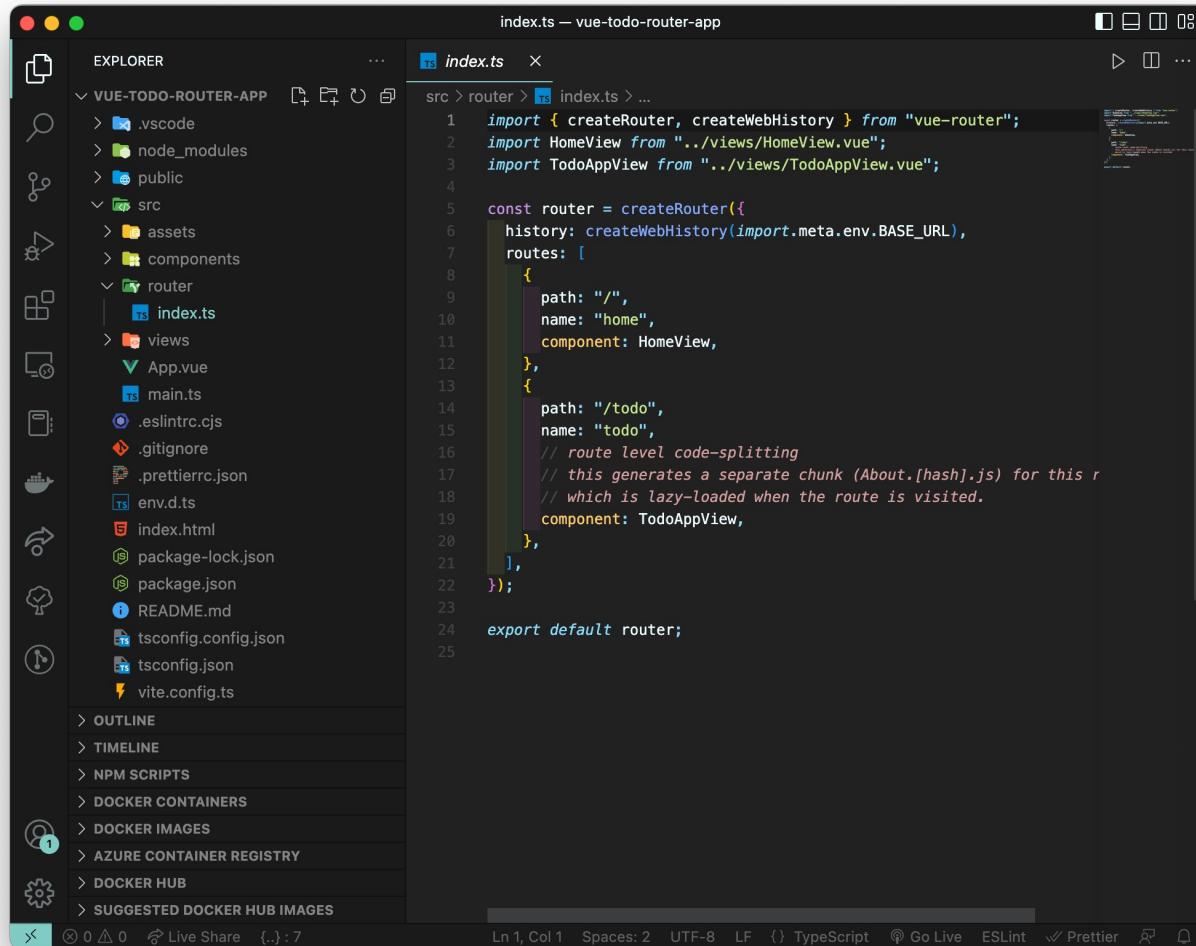
<style scoped>
header {
  line-height: 1.5;
  max-height: 100vh;
}

```

The "EXPLORER" sidebar shows the project structure, including "src" with "index.ts", "router" with "index.ts", "views" with "App.vue", "main.ts", and "TodoAppView.vue". The "ROUTER" section of the Explorer shows routes: Home and Todo.

The bottom status bar shows the following icons and text: "0 △ 0" (linter), "Live Share" (with a "1" badge), "Vue", "ESLint", "1 known issue" (highlighted in orange), "tsconfig.json", "<TagName prop-name /> 4.9.4 (vue)", "Prettier", and a bell icon.

# With Router



index.ts — vue-todo-router-app

```
1 import { createRouter, createWebHistory } from "vue-router";
2 import HomeView from "../views/HomeView.vue";
3 import TodoAppView from "../views/TodoAppView.vue";
4
5 const router = createRouter({
6   history: createWebHistory(import.meta.env.BASE_URL),
7   routes: [
8     {
9       path: "/",
10      name: "home",
11      component: HomeView,
12    },
13    {
14      path: "/todo",
15      name: "todo",
16      // route level code-splitting
17      // this generates a separate chunk (About.[hash].js) for this r
18      // which is lazy-loaded when the route is visited.
19      component: TodoAppView,
20    },
21  ],
22});
23
24 export default router;
25
```

EXPLORER

- VUE-TODO-ROUTER-APP
  - .vscode
  - node\_modules
  - public
  - src
    - assets
    - components
    - router
      - index.ts
    - views
      - App.vue
      - main.ts
  - eslintrc.cjs
  - .gitignore
  - .prettierrc.json
  - env.d.ts
  - index.html
  - package-lock.json
  - package.json
  - README.md
  - tsconfig.config.json
  - tsconfig.json
  - vite.config.ts

OUTLINE

TIMELINE

NPM SCRIPTS

DOCKER CONTAINERS

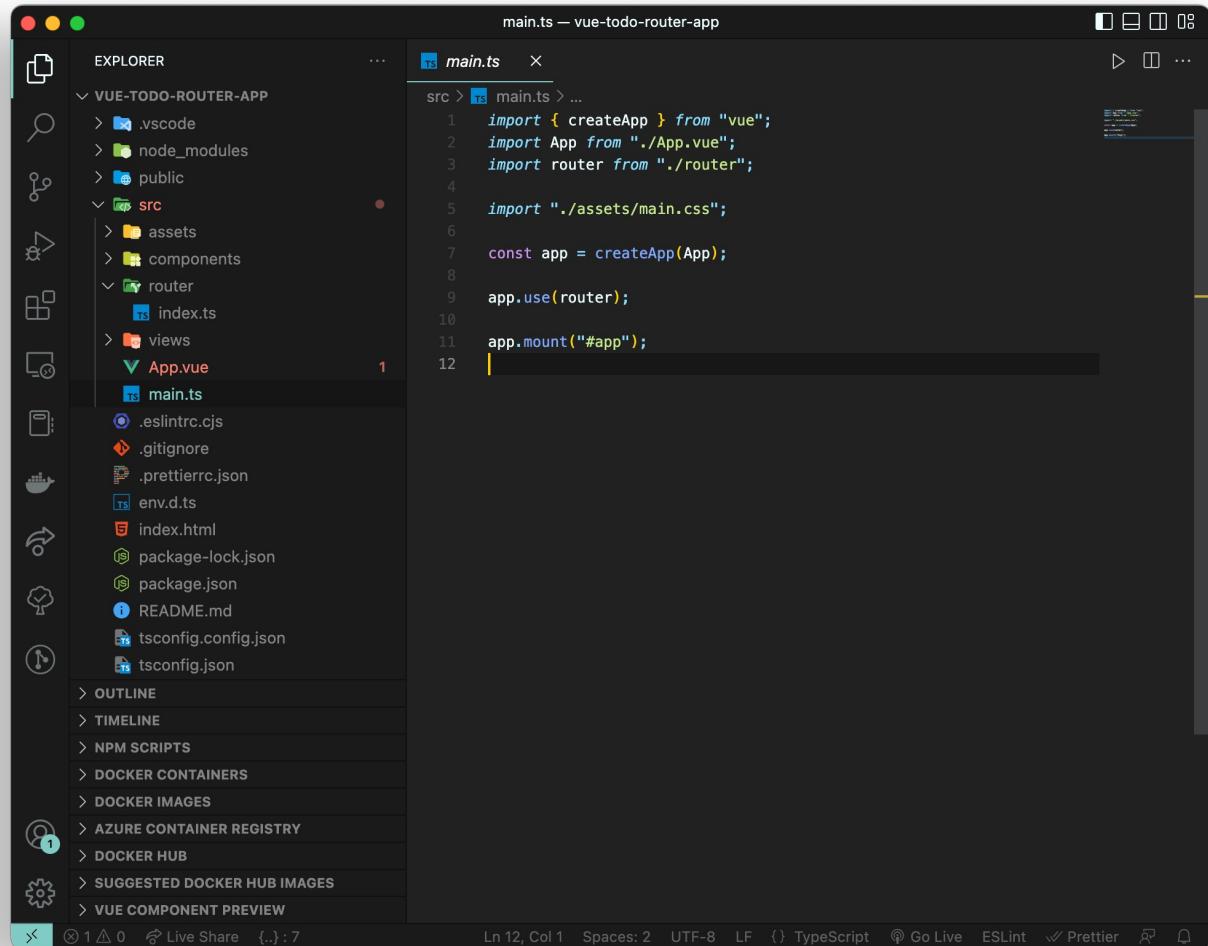
DOCKER IMAGES

AZURE CONTAINER REGISTRY

DOCKER HUB

SUGGESTED DOCKER HUB IMAGES

Ln 1, Col 1 Spaces: 2 UTF-8 LF () TypeScript ⚡ Go Live ESLint ✅ Prettier 🔍 🔔



main.ts — vue-todo-router-app

```
1 import { createApp } from "vue";
2 import App from "./App.vue";
3 import router from "./router";
4
5 import "./assets/main.css";
6
7 const app = createApp(App);
8
9 app.use(router);
10
11 app.mount("#app");
12
```

EXPLORER

- VUE-TODO-ROUTER-APP
  - .vscode
  - node\_modules
  - public
  - src
    - assets
    - components
    - router
      - index.ts
    - views
      - App.vue
      - main.ts
  - eslintrc.cjs
  - .gitignore
  - .prettierrc.json
  - env.d.ts
  - index.html
  - package-lock.json
  - package.json
  - README.md
  - tsconfig.config.json
  - tsconfig.json

OUTLINE

TIMELINE

NPM SCRIPTS

DOCKER CONTAINERS

DOCKER IMAGES

AZURE CONTAINER REGISTRY

DOCKER HUB

SUGGESTED DOCKER HUB IMAGES

VUE COMPONENT PREVIEW

Ln 12, Col 1 Spaces: 2 UTF-8 LF () TypeScript ⚡ Go Live ESLint ✅ Prettier 🔍 🔔

## With Router

```
"dependencies": {  
    "vue": "^3.2.45",  
    "vue-router": "^4.1.6"  
},
```

```
private": true,  
"dependencies": {  
    "@angular/animations": "^15.0.0",  
    "@angular/common": "^15.0.0",  
    "@angular/compiler": "^15.0.0",  
    "@angular/core": "^15.0.0",  
    "@angular/forms": "^15.0.0",  
    "@angular/platform-browser": "^15.0.0",  
    "@angular/platform-browser-dynamic": "^15.0.0",  
    "@angular/router": "^15.0.0",  
    "rxjs": "~7.5.0",      Bong, 6 days ago • Chore: 리액트 및  
    "tslib": "^2.3.0",  
    "zone.js": "~0.12.0"  
},
```

# V 개발 환경 설정

```
kanghyun@KangHyun-MacBook:~/projects/vue-todo-app ✘ 20:29:27
~/projects ➤ cd vue-todo-app
~/p/vue-todo-app ➤ npm install
✓ 20:29:31

up to date, audited 282 packages in 442ms

84 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
~/p/vue-todo-app ➤ npm run lint
✓ 20:29:34

> vue-todo-app@0.0.0 lint
> eslint . --ext .vue,.js,.jsx,.cjs,.mjs,.ts,.tsx,.cts,.mts --fix --ignore-path
.gitignore

~/p/vue-todo-app ➤ npm run dev
✓ 20:29:41

> vue-todo-app@0.0.0 dev
> vite

VITE v4.0.4 ready in 514 ms

→ Local: http://localhost:5173/
```



# Volar 설치 (VSCode Extension)

The screenshot shows the VSCode Extension Marketplace. On the left, a sidebar lists various extensions related to Vue, including 'Vue Language Features (Volar)', 'Vue', 'Syntax Highlight for Vue.js', 'Vue 3 Snippets', 'Vue VSCode Snippets', 'Vue Theme', 'Vue Peek', 'TypeScript Vue Plugin (Volar)', 'vue-helper', 'Vue VS Code Snippets', and 'vue-format'. The main panel displays the details for the 'Vue Language Features (Volar)' extension, version v1.0.24. It has a rating of 4.5 stars from 80 reviews. The description states: 'Language support for Vue 3'. The 'Details' tab is selected, showing sections for 'Vue Language Features', 'Quick Start' (with items like 'create-vue', 'Vitesse', 'petite', 'vue3-eslint-stylelint-demo', and 'volar-starter'), 'Usage' (with items like 'Setup for Vue 2' and 'Define Global Components'), and 'Notes'. The 'Categories' section includes 'Programming Languages'. The 'Extension Resources' section links to the Marketplace, Repository, License, and Vue pages. The 'More Info' section provides publication and update details.

구문 강조 표시, typescript 지원, 템플릿 표현식 및 컴포넌트 지원

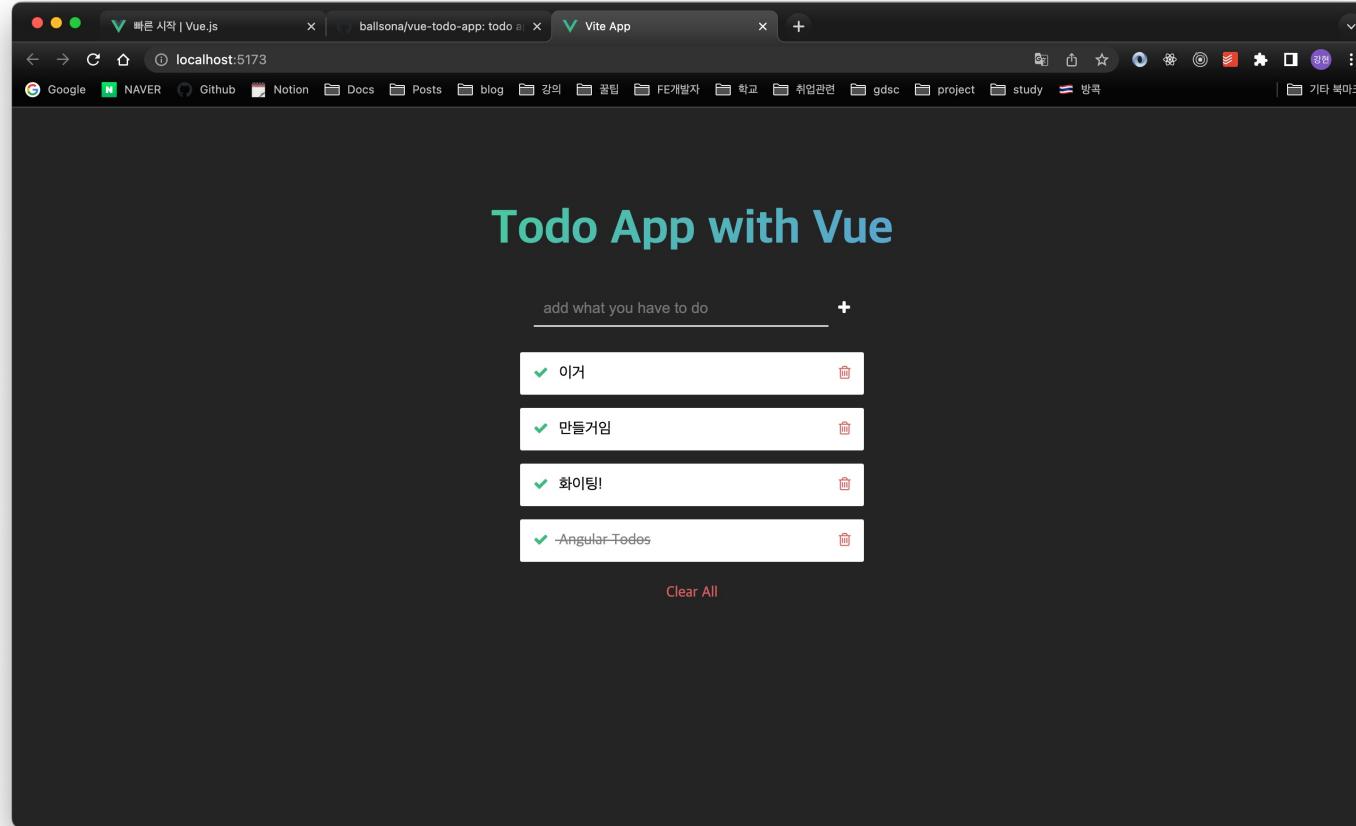


# Awesome-icon 추가 및 불필요한 파일 삭제

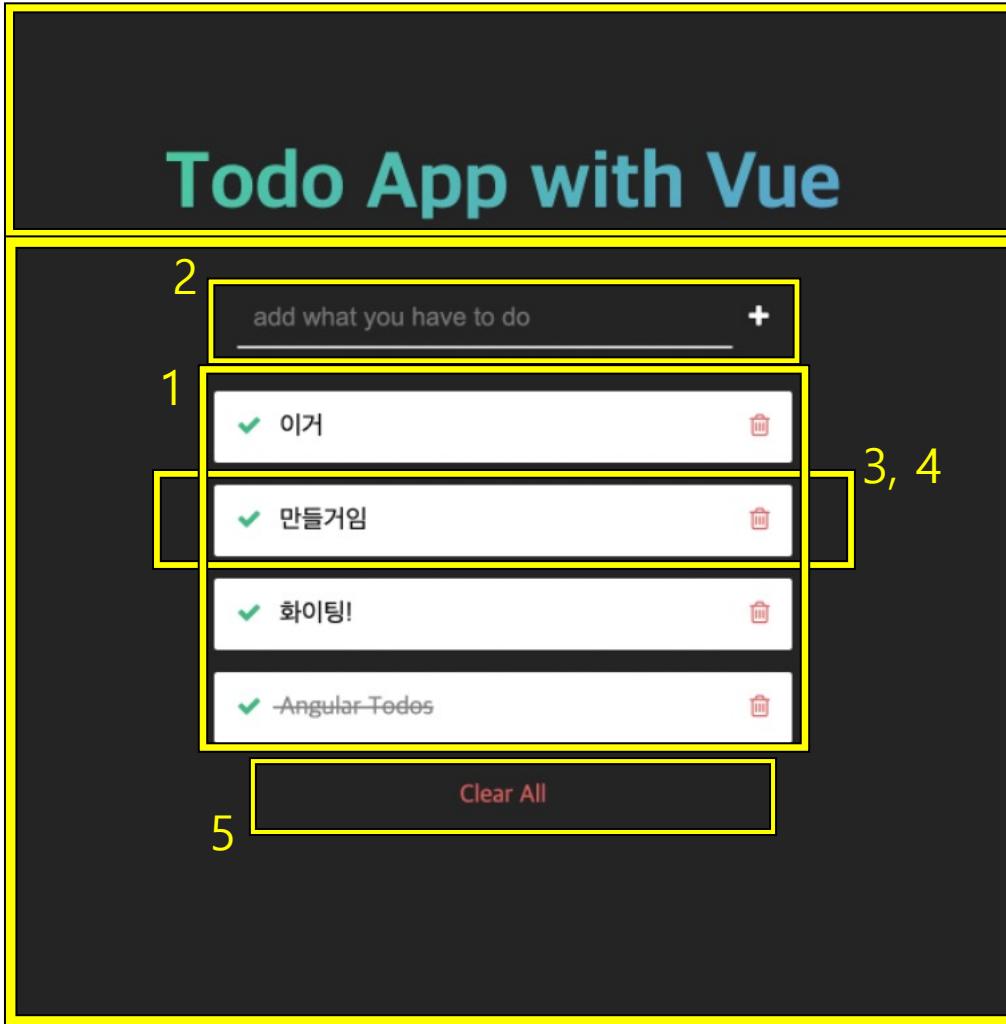
The screenshot shows the VS Code interface with the following details:

- EXPLORER:** Shows the project structure for "VUE-TODO-APP".
- index.html:** The current file being edited.
- Content:** The code for the index.html file includes the following relevant parts:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <link rel="icon" href="/favicon.ico">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Vite App</title>
  </head>
  <body>
    <div id="app"></div>
    <script type="module" src="/src/main.ts"></script>
  </body>
</html>
```
- Bottom Status Bar:** Shows "Ln 6, Col 1 (128 selected) Spaces: 2 UTF-8 LF HTML ⚡ Go Live ESLint ✨ Prettier" and icons for Live Share and Docker.

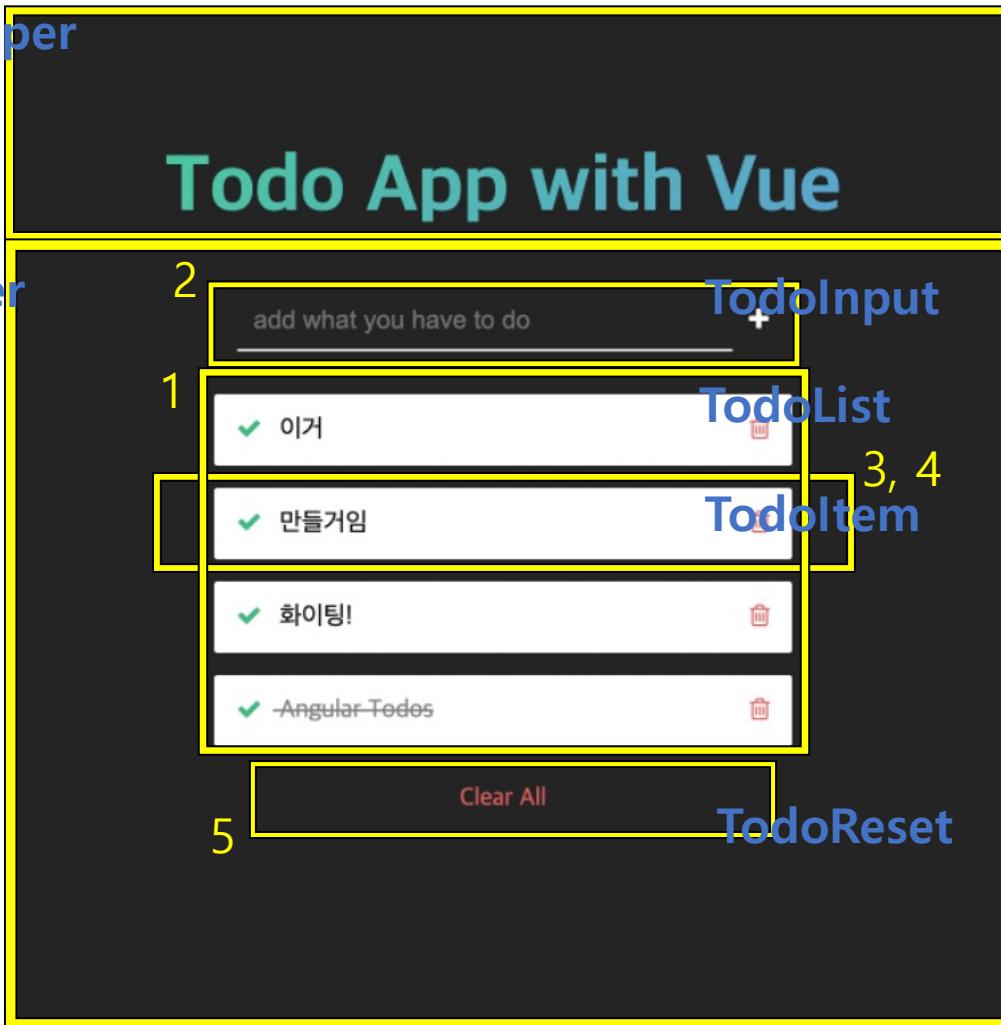


1. Todo 전체 보여주기
2. Todo 추가
3. Todo 상태 토글
4. Todo 삭제
5. Todo 전체 삭제
6. 저장소 저장 (LocalStorage)



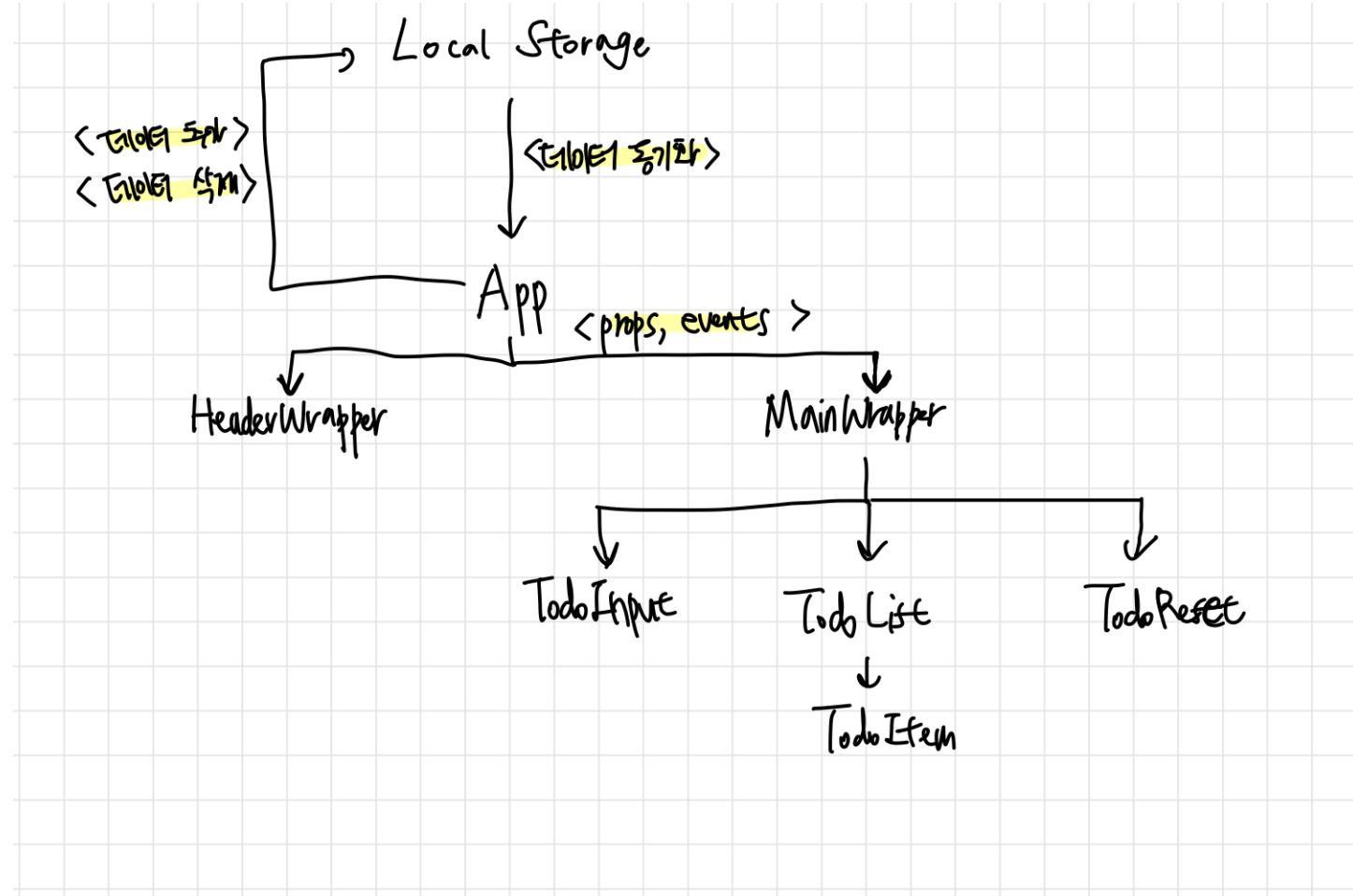
1. Todo 전체 보여주기
2. Todo 추가
3. Todo 상태 토글
4. Todo 삭제
5. Todo 전체 삭제
6. 저장소 저장

## HeaderWrapper



- App
  - HeaderWrapper
  - MainWrapper
    - TodoInput
    - TodoList
      - TodoItem
    - TodoReset

# ▼ 애플리케이션 구조



A screenshot of the Visual Studio Code (VS Code) interface. The title bar shows "storage.ts — vue-todo-kanghyun-app". The left sidebar (EXPLORER) displays the project structure:

- src
- assets
- components
  - HeaderWrapper.vue
  - MainWrapper.vue
  - Todolinput.vue
  - Todoltem.vue
  - TodoList.vue
  - TodoReset.vue
- utils
  - storage.ts

The main editor area shows the content of storage.ts:

```
You, 3 hours ago | 1 author (You)
1 export const callData = (key: string) => {
2   const savedData = localStorage.getItem(key);
3   return savedData && JSON.parse(savedData);
4 };
5
6 export const saveData = (value: unknown, key: string) => {
7   const toJson = JSON.stringify(value);
8   localStorage.setItem(key, toJson);
9 };
10
```

At the bottom, the status bar shows: Spaces: 2, UTF-8, LF, {}, TypeScript, Go Live, ESLint, Prettier.

LocalStorage에 저장!

경로) src/utils/storage.ts

# App.vue

# HeaderWrapper.vue

# MainWrapper.vue

# TodoInput.vue

# TodoList.vue

# Todoltem.vue

# **TodoReset.vue**

# 스타일 코드 분리?

## 질문 및 피드백

# Vue vs React

人門